CLAIMS

What is claimed is:

1. A method comprising:

maintaining first and second return buffers for first and second instruction pipelines respectively, both buffers having a plurality of return address entries; and

sending a request to the second instruction pipeline to fill the first return buffer with the entries in the second return buffer.

2. The method of claim 1, further comprising detecting a return instruction prior to sending a request to the second instruction pipeline to fill the first return buffer with the entries in the second return buffer.

3. The method of claim 2, further comprising popping the first return buffer after detecting a return instruction.

4. The method of claim 2, further comprising determining if the return instruction was detected by a third instruction pipeline and sending a return address in the first return buffer to the third instruction pipeline if the return instruction was not detected by the third instruction pipeline.

5. The method of claim 3, further comprising updating a pointer to the first return buffer after popping the first return buffer.

6. The method of claim 2, further comprising detecting a call instruction prior to detecting a return instruction.

7. The method of claim 6, further comprising pushing a return address corresponding to the call instruction onto the first return buffer after detecting a call instruction.

8. The method of claim 7, further comprising updating a pointer to the first return buffer after pushing the return address onto the first return buffer.

9. The method of claim 8, further comprising determining if the call instruction was detected by a third instruction pipeline and sending a return address in the first return buffer to the third instruction pipeline if the call instruction was not detected by the third instruction pipeline.

10. The method of claim 4, wherein determining if the return instruction was detected by a third instruction pipeline comprises determining if the return instruction was detected by a pipeline for predicting whether branches will be taken.

11. The method of claim 1, wherein maintaining a first return buffer for a first instruction pipeline comprises maintaining a first return buffer for a pipeline to translate instructions fetched from a memory device.

12. The method of claim 1, wherein maintaining a second return buffer for a second instruction pipeline comprises maintaining a second return buffer for a pipeline that predicts branches in a cache.

13. The method of claim 7, wherein each entry in the first return buffer comprises a valid bit that indicates whether the corresponding return address is valid.

14. The method of claim 13, further comprising checking the valid bit corresponding to each popped return address to determine if the return address is valid.

15. The method of claim 1, wherein the first and second return buffers are return stacks.

16. An instruction pipeline in a microprocessor comprising:

a decode unit, the decode unit to maintain a first return buffer having a plurality of return address entries; and

a branch prediction unit, the branch prediction unit to maintain a second return buffer and to fill the first return buffer with entries from the second return buffer.

17. The instruction pipeline of claim 16, wherein the decode unit to further recognize call instructions and return instructions.

18. The instruction pipeline of claim 17, wherein the branch prediction unit to fill the first return buffer with entries from the second return buffer comprises the branch prediction unit to fill the first return buffer with entries from the second return buffer each time the decode unit recognizes a return instruction.

19. The instruction pipeline of claim 16, wherein decode unit to further send a request to the branch prediction unit to fill the first return buffer with entries from the second return buffer each time the first return buffer is not full.

20. The instruction pipeline of claim 17, further comprising a second branch prediction unit, the decode unit to send the second branch prediction unit a return

address in the first return buffer if the decode unit recognizes a call instruction or a return instruction that the second branch prediction unit did not recognize.

21. The instruction pipeline of claim 16, wherein the branch prediction unit comprises a cache branch prediction unit to predict branches in a cache.

22. The instruction pipeline of claim 21, wherein the cache branch prediction unit to notify the decode unit when there is a cache miss and the decode unit to send a request to the cache branch prediction unit to fill the first return buffer with entries from the second return buffer.

23. A method comprising:

maintaining a return buffer for a decode unit, the return buffer having a plurality of return address entries;

detecting a call instruction;

determining if the call instruction was detected by a branch prediction unit; and

sending a return address to the branch prediction unit if the call instruction was not detected by the branch prediction unit.

24. The method of claim 23, further comprising pushing a return address corresponding to the call instruction onto the return buffer.

25. The method of claim 23, further comprising detecting a return instruction.

26. The method of claim 25, further comprising popping a return address from the return stack buffer.

27. The method of claim 26, wherein each entry in the return buffer comprises a valid bit that indicates whether the return address is valid.

28. The method of claim 27, further comprising checking the valid bit corresponding to the popped return address to determine if the return address is valid.

29. The method of claim 23, further comprising sending a request to a cache branch prediction unit to fill the return buffer.

30. The method of claim 25, further comprising determining if the return instruction was detected by the branch prediction unit and sending a return address in the return buffer to the branch prediction unit if the return instruction was not detected by the branch prediction unit.